Thermal and electro-thermal modeling and simulation techniques for multichip modules

Technical report

SPC 98-4023 Project

Marta Kerecsen Rencz and Vladimir Székely
Technical University Budapest

Budapest, Hungary
12.10.1998

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 10 December 1998	3. REPORT TYPE AND DATES COVERED Final Report

4. TITLE AND SUBTITLE

Thermal and Electro-Thermal Modeling and Simulation Techniques for Multichip Modules.

5. FUNDING NUMBERS

F61775-98-WE040

6. AUTHOR(S)

Dr. Istvanne Kerecsen

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Technical University of Budapest Department of Electron Devices
Goldmann ter 3
Budapest H-1521
Hungary

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

EOARD
PSC 802 BOX 14
FPO 09499-0200

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

SPC 98-4023

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

A

13. ABSTRACT (Maximum 200 words)

This report results from a contract tasking Technical University of Budapest Department of Electron Devices as follows: The contractor will investigate the issues and problems of coupled electro-thermal M&S of Mixed Technology Multichip Modules (MT-MCMs). MT-MCMs for this project will be defined as a mixtures of high performance computing, microelectromechanical systems, and microfluidic systems. The use of M&S will be investigated for each of the generic steps of the design process flow as shown in Figure 1 of the proposal. The contractor will determine the most cost effective M&S approach for each step in the design process. The possibilities of integrating the existing fast and accurate design tools of TUB into the AFRL design flow have to be investigated, and the optimal M&S approach determined.

14. SUBJECT TERMS

EOARD, Modelling & Simulation, Microelectronics, MEMs, multichip modules, mixed technology

15. NUMBER OF PAGES

37

16. PRICE CODE

N/A

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19, SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

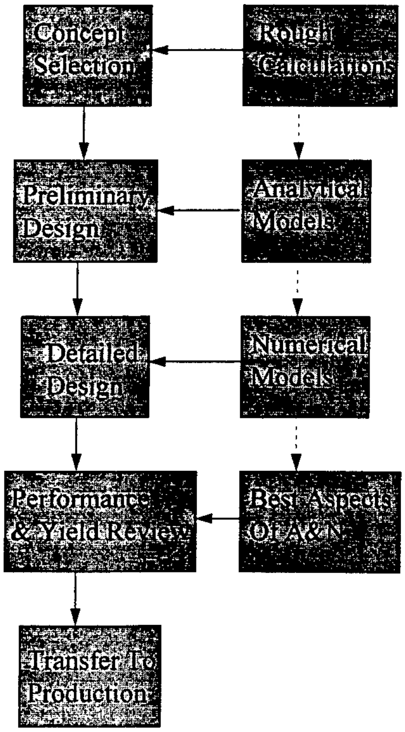
Contents

Thermal and electro-thermal modeling and simulation techniques for multichip modules

1. Introduction

1.1 The purposes of the work

The purpose of our activity is to study and elaborate suitable strategies for the thermal simulation and verification of MCM designs in the design flow of Fig.1.



- Fig.1. The considered design flow

The main questions to be investigated are as follows:

- Where is the optimal place of the thermal or coupled electro-thermal simulation in the design flow?

- Which are the optimal tools for thermal and coupled electro-thermal simulation – taking into account the accuracy/computer time trade-off and the real requirements in accuracy, and the availability and feasibility of such tools?
- What are the problems that have to be simulated thermally and whether the steady-state thermal simulation is sufficient? How can the scope of thermal simulation be extended to the dynamic problems?
- Is it possible to simplify the thermal and electro-thermal simulation or design verification by using simplified, compact thermal models of the MCM module? How can such compact models be generated?

The second purpose of the work is to investigate how the access of alternative thermal solvers can be provided in a design environment.

Having a choice of thermal solvers in the framework, the following advantages can be identified:

- For each problem the most suitable solver can be used, e.g. a rough simulation at higher levels, while accurate but rather time consuming simulation at lower levels, etc.
- More accurate or faster solution modules can be used at certain levels of the design flow instead of the usually inaccurate and time-consuming FEM codes.
- Frequency or time domain dynamic solution, thermal model generation, etc. becomes also possible with the suggested new tools.
- The different solvers can be compared, the resolution, accuracy etc. features can be investigated and matched to the given problem, etc.

1.2.The Benchmark MT-MCM Model

The benchmark MT-MCM Model used throughout this report is such a multichip module, which contains high performance computing chips, micro-electro-mechanical systems (MEMS), and possibly microfluidic systems.

2. Strategy of the thermal verification during the MCM design

The strategy of the **thermal** simulation/verification in the design flow should be influenced strongly by the fact that the thermal simulation may require huge amount of computer time – especially if high-resolution thermal map is needed or if dynamic (transient) solution is calculated. So, a great care should be applied deciding where and how coupled or thermal simulation steps have to be inserted into the design flow. Both the long waiting times of superfluous thermal simulations and the superficial handling of the thermal issues should be eliminated at the same time.

The strategy for the **coupled electro-thermal** simulation requires a bit more care. *Coupled* means that there is a two-way interaction between the electrical and the thermal subsystems. The electronic circuit dissipates heat therefore acts as a source for the thermal part. Inversely, the temperature response of the thermal part influences the operation of the electrical part. In order to follow these phenomena, an elegant and theoretically well established method is frequently used: namely the *simultaneous solution* of the electrical and thermal subsystems. This method, however, requires the modification of the code of both the electrical and the thermal solvers. Moreover, if the complexity of the electrical circuitry is beyond the limit of handling it on component level, the simultaneous solution becomes to be problematic. In such cases coupling of the simulators is the appropriate way. This means that the electrical and the thermal solutions do not occur simultaneously but alternately. In our opinion this is the right way for the coupled electro-thermal simulation in a design environment where at least one of the following criteria is fulfilled:

- Higher level simulation tools are used as well (e.g. logic gate level, register transfer level or behavioral level),
- Easy linking of new simulation tools is a requirement, without the need of any modification in the simulator code,
- Not solely the pure electrical simulators but electromagnetic and other solvers are of concern.

In our opinion the following rules should be kept in mind in order to find the right place(s) of thermal or electro-thermal simulation step(s) in the design flow:

- A fast thermal simulation (even with moderate accuracy) should be applied as early as possible, in order to highlight the rough thermal problems in the early phase of the preliminary design. Such serious problems can be corrected usually by the rearrangement of the chip placement but sometimes only by choosing other package and/or cooling scheme. The early thermal analysis helps us to avoid useless investment of work into a design that is thermally unmanageable.

- A detailed and accurate thermal simulation is needed at the final phase of the detailed design. This simulation helps us to produce the thermal specification of the MCM module, including the thermal resistance towards the ambience, the maximum values of the chip temperatures, etc. If strong interactions can be expected in the system, at this point electro-thermal simulation is needed, otherwise the purely thermal solution is sufficient. The dissipation values should be calculated however properly from the data of the detailed electrical simulation even in the latter case.

- In cases when the MCM unit operates in transient mode thermal or electro-thermal transient simulation is also needed.

- In the case when the thermal interactions between the chips of the MCM are considerable and may influence the overall quality of operation, the models of the internal thermal couplings are also needed. Compact models for an MCM package can be generated also from the results of the thermal simulation. The advantage of these compact models manifests in the fact that if such a model has been derived once, it is possible to predict the thermal behavior of the package, without further time-consuming field-solver simulations. In a number of cases the coupled electro-thermal simulation uses these compact models as well. Compact thermal models are indispensable if fluid flow is present in the system.

- The thermal effect of fluid flow has to be simulated by CFD (Computational Fluid Dynamics) programs. These programs request usually extremely long run times (for usual systems this can be in the order of magnitude of several weeks on supercomputers), so the simplicity of the models of the system elements is a key issue.

Conclusion

We propose to use a *fast thermal solver in the early phase of the design*, in the first moment when

(i) the package has been chosen,

(ii) the first placement of the chips is done,

(iii) the dissipation of the chips is known (or estimated).

We propose to use a *high-resolution and accurate thermal solver in the final phase* of the design, in conjunction with the precise dissipation and boundary condition data. In this phase the electro-thermal simulation is also needed if strong interactions are expected. This simulation will give the thermal specification data of the MCM. Beside this, based on this simulation, the compact thermal model of the design can be generated as well.

3. The program modules proposed to be linked to a general purpose design framework

3.1. Thermal field solver based on the Fourier method (THERMAN)

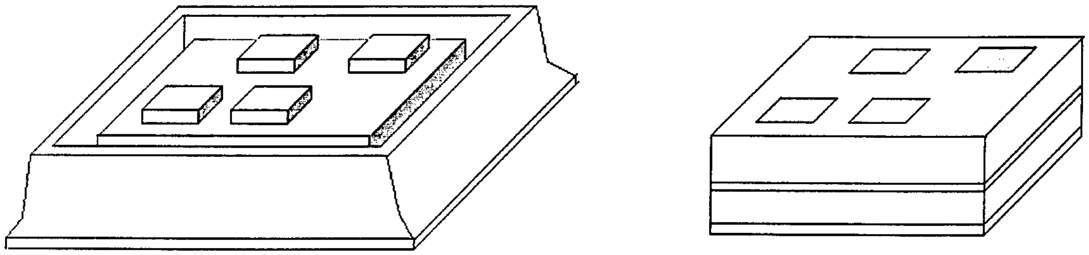
This program module is able to calculate the thermal field in rectangular shape, multi-layer structures. The physical structure of the lateral MCMs is rather close to this model. This allows very fast calculation of the thermal field, although some inherent constraints of the model limit the accuracy of the results.

The detailed features of this solution engine are the followings:

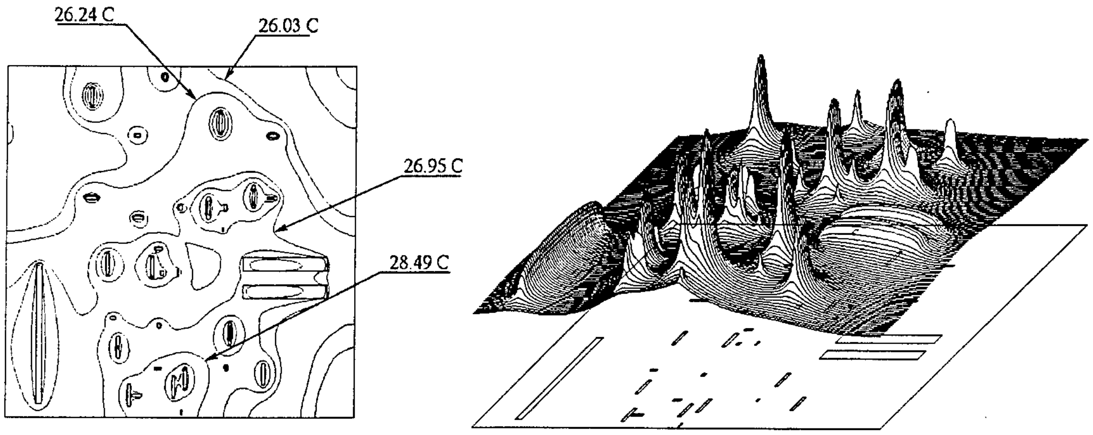
- Number of the different layers: unlimited
- Resolution on the surface: from 256x256 to 2048x2048
- Boundary conditions, top surface and side walls: adiabatic, isothermal or natural convection
- Boundary conditions, bottom surface: isotherm or described by a 2D thermal impedance matrix
- Heat removing through leads or upper-side heat-sinking pistons can be taken into account
- Solutions: steady-state and frequency-domain
- Algorithm: Fourier method, realized by using FFT
- Constraints: the layers must be of equal size (but may have different thickness values), the thermal parameters of the different materials are assumed to be constant (not depending on the temperature), the thickness of the chips inserted into the MCM can not be taken into account.

In order to visualize the used model a real MCM structure and his thermal model is demonstrated in Fig.2. The different layers of the MCM package appear in the model as well. Note however, that this tool does not model possible differences in the lateral layer sizes. The dissipating chips appear on the surface of the model with their real position and sizes. The thickness of the chips is neglected – they appear as 2D dissipating areas.

The results of the analysis can be plotted e.g. by using a set of isothermal lines or by using pseudo-color maps. Two examples of such thermal map plots are shown in Fig.3.



• Fig.2. MCM package and its thermal model in the *THERMAN* program



• Fig.3. Plots of thermal maps calculated by the *THERMAN* program

The detailed description of the model and the solution algorithm of this program can be found in Appendix 1.

3.2. Thermal field solver based on the finite differences and successive node reduction (SUNRED)

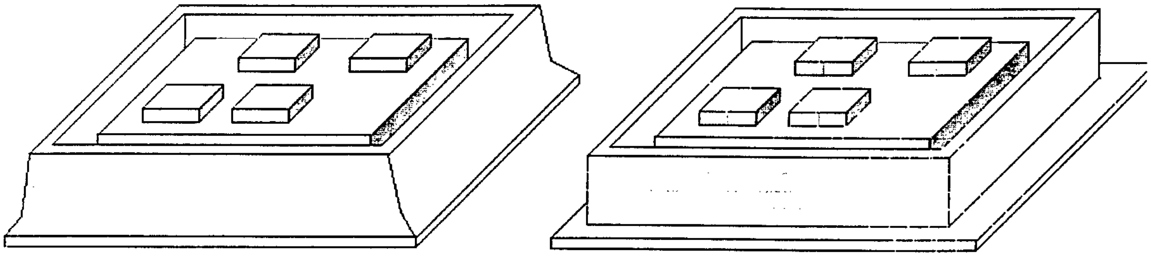
This program module is based on the finite difference model of the investigated structure. The structure is mapped into a lumped RC network. This network is solved using an original method called ***SU****ccessive* ***N****etwork* ***RED****uction* algorithm. This algorithm provides an acceptable solution time even for model networks having a great number of the network nodes. The solver has been already tested to handle 2D problems appropriately. The extension for 3D problems is just in the finishing phase. The model can follow the real structure of the MCM module including the thickness of the chips and the different lateral sizes of the layers of the package.

The expected features of the 3D version of this simulation tool are the followings:

- Lateral resolution: from 32x32 to 128x128
- Vertical resolution: from 4 to 16
- Grid: variable size
- Boundary conditions: arbitrary. Natural convection can be modeled as well.
- Solutions: steady-state and time-domain (transient)
- Algorithm: finite difference, successive network reduction
- Constraints: Linear solution (the thermal parameters of the different materials does not depend on the temperature)

A real MCM structure and its SUNRED model are plotted in Fig.4.

The detailed description of the model and the solution algorithm of this program can be found in Appendix 2. This description concerns the 2D tool but the extension to the 3D problems is straightforward.



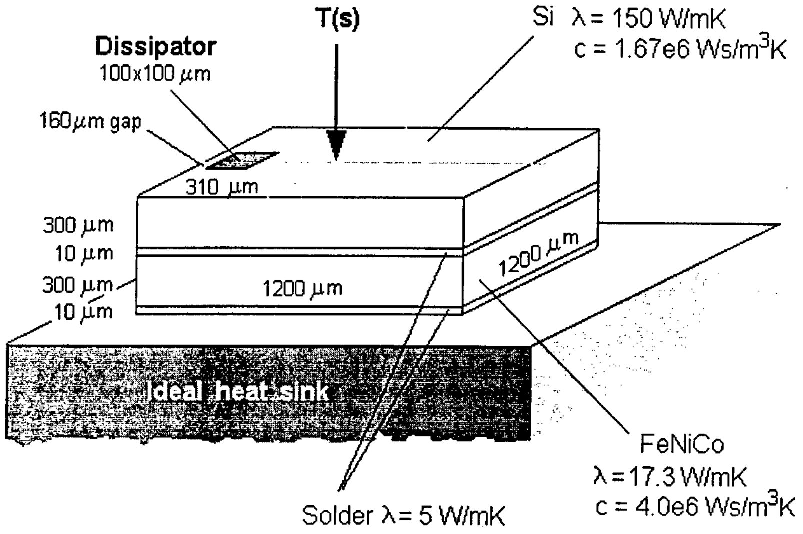
- Fig.4. An MCM structure and the corresponding *SUNRED* model

3.3. Thermal compact model generator (THERMODEL)

The use of the common thermal simulation tools (as FEM codes, Fourier solvers etc.) is rather time consuming, so that it is highly advisable to avoid repeated runs necessary during the design process. Using the results of one single 3D thermal simulation run a simplified compact thermal model can be extracted. This latter can be used in the subsequent simulations leading to a significant save in the run-time.

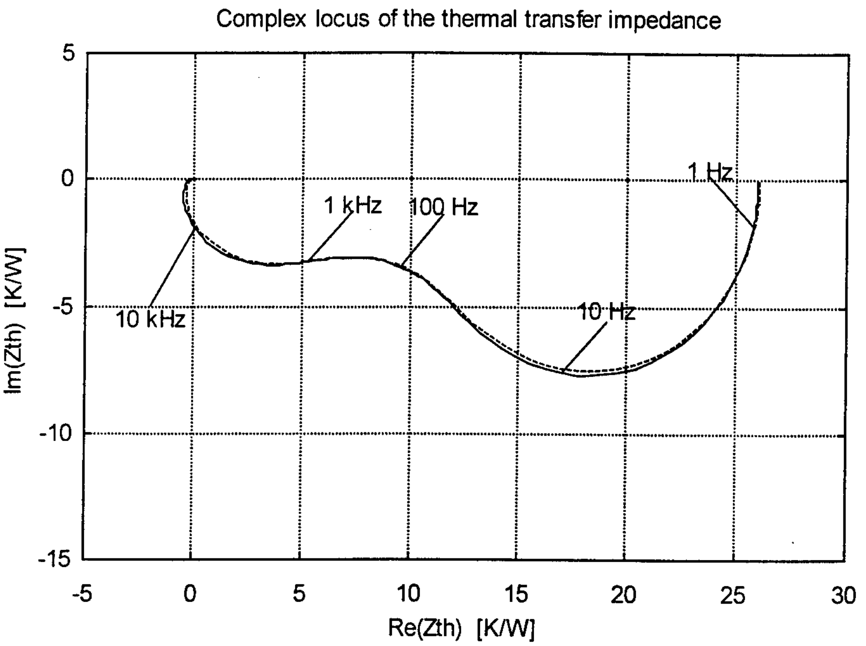
THERMODEL is a software tool, which generates compact thermal models to describe the heat conduction of 3D physical structures, from either the time-domain or the frequency-domain response of the given structure. The generated compact model consists of a single (or twin) RC ladder of eight-twelve stages. Both the one-port and the transfer thermal behavior can be modeled. These compact models can be useful in circuit-level simulators (like SPICE or ELDO) where thermal effects have to be considered in the models or in the electrothermal IC chip simulators where all the relevant thermal couplings of the chip can be modeled in this way.

In the next example a conventional IC structure will be investigated. The operation of high-gain monolithic operational amplifiers can be heavily disturbed by the thermal feedback from the output transistors toward the input stage. Let the problem be to model this feedback path for a real structure.



- Fig.5. Physical outlines of an IC chip. The thermal transfer impedance has been calculated between the Dissipator area and the point T(s).

The chip arrangement and the main dimensions are shown in Fig.5. The THERMAN program calculated the heat distribution on the chip as function of the frequency. The 3D-field calculation provides the Bode plot or the complex locus of the thermal transfer impedance. The identification will be carried out in the frequency domain in this case.



• Fig.6. Responses of the IC chip. Solid line: calculated by the 3D field solver, dashed line: response of the generated model

The complex locus of the thermal transfer impedance calculated by the THERMAN program is shown in Fig.6. (solid line). The THERMODEL program provides the time-constant spectrum as the result of the identification. The model network generated by THERMODEL consists of 17 stages.

4. The problems to be solved in order to link the new knowledge sources

4.1. The required input data

In order to build the model of the investigated MCM structure the following groups of data are needed:

- files for the thermal material parameters (heat conductivity, thermal capacitance),
- description of the MCM package structure (geometrical sizes, etc.),
- description of the chips mounted into the MCM (sizes, position),
- power map of the chips (or, at least, the list of the integral heat dissipation of the individual chips).

4.2. The required control data

Each new solver module requires a few control data, like the resolution, the steady state/dynamic flag, time interval definition in the dynamic-transient case, etc. Following the conception of the Blackboard Framework we propose to collect these control data in an (alphanumeric) file. This file can be generated by the control mechanism of the framework or by using a command-window with radio buttons .

4.3. Results provided for the further evaluation and for display

The results are also planned being written into files of the framework as well. Two alternatives are open when defining the format of these results:

- to follow the format already used in the Framework for thermal analysis results
- to define a format more appropriate for the new simulation engines.

In the latter case a set of further program modules is needed, which are capable to translate these data into the format required by the other system modules (as displaying results in graphics windows etc.)

The results of the compact thermal model identification will be generated in the format of the standard network netlists (as e.g. the format of SPICE netlists).

References

[1] V. Székely, A. Csendes, M. Rencz: μS-THERMANAL: an efficient thermal simulation tool for microsystem elements and MCM's, *SPIE'96 Symposium on Micromachining and Microfabrication,* 14-15 Oct. 1996, Austin, Texas USA, SPIE Proc. V.2880, pp.64-75.

[2] V. Széķely, M. Rencz: Fast field solver programs for thermal and electrostatic analysis of microsystem elements, *Proc. of ICCAD*, San Jose, CA, Nov. 1997, pp 684-689

[3] V. Székely, M. Rencz: Thermal Investigation of Microsystems, Multichip Modules with integrated Sensors. In *NATO ASI Series,* 3 HT-Vol. 16, Kluwer Acad. Publ. pp 1-10, 1996

[4] V. Székely, A. Poppe, M. Rencz, A. Csendes, A. Páhi: Self-Consistent Electro-Thermal Simulation: Fundamentals and Practice ", *Microelectronics Journal* V28, No 3, pp. 247-263

[5] V. Székely, A. Páhi, A. Poppe, M. Rencz, A. Csendes: SISSI - a tool for dynamic electro-thermal simulation of analog VLSI cells, *European design and Test Conference'97*, 17-20 March 1997, Paris, France, p617

[6] V. Székely, M. Rencz, B. Courtois: Thermal investigations of IC's and Microstructures, *Encyclopedia of Microcomputer Series,* Marcel Dekker, 1998

[7] V. Székely, A. Poppe, A. Páhi, A. Csendes, G. Hajas, M. Rencz: Electro-thermal and logi-thermal simulation of VLSI designs. *IEEE Trans. on VLSI Systems*, Vol.5, No.3, pp 258-269, 1997

[8] V. Székely, M. Rencz, B. Courtois: CAD tools for thermal testing of electronics systems, *Advances in Electronic Packaging,* 1997, EEP-Vol.19-2,pp.2209-2215

[9] V. Székely, M. Rencz, B. Courtois: Advances in tracing he thermal behavior if IC, *IEEE Design and Test of Computers,* April-June 1998, pp. 14-21

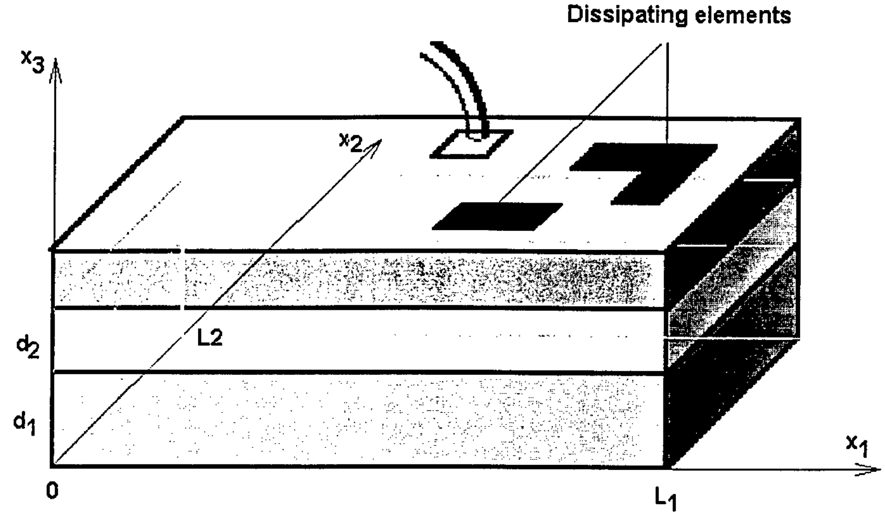
APPENDICES

Appendix 1.

Algorithm of the THERMAN program

A1.1. The Fourier algorithm for multi-layered structures

Our research team has been involved in the thermal simulation of IC chips for 18 years. The simulation tool THERMANAL developed at our department is based on th Fourier method. The reason of this choice is the obtainable very quick computation compared to the other solutions such as the FEM or the finite difference methods. Improvements developed by our group, such as taking into account the non-ideal nature of the heat-sink, calculating the heat distribution of beam-lead like structures etc. were already reported a decade ago.



• Fig. A1.1. The model of the IC chip

Let us first discuss the "classic" multilayer solution. The considered structure is shown in Fig.A1.1. Equally-shaped rectangular layers are stacked on an ideal heat sink. Each layer is characterized by its specific thickness d_i , heat conductivity λ_i and unit volume heat capacitance c_i. The dissipating elements are lying on the upper surface of the uppermost layer. Heat removal is accomplished only on the bottom surface of the structure while the "sidewalls" are adiabatic. Heat transfer is assumed only by conduction. This is a reasonably good model for a conventional IC chip. The heat-flow differential equation can be written as

$$divgrad\, T = \frac{c}{\lambda}\frac{\partial T}{\partial t} \qquad \text{(A1.1)}$$

for a homogeneous medium with constant λ heat conductivity and c unit-volume heat capacitance. Having more layers with different λ_i and c_i values this equation is applied for each region and are matched on the layer interfaces.
The boundary conditions to be fulfilled by the overall solution are:

$$\left.\frac{\partial T}{\partial x_i}\right|_{x_i=0.L_i} = 0 \qquad \text{i = 1,2} \qquad \text{(A1.2)}$$

for the four "sidewalls" of the multi-layered parallelepiped (where L_1 and L_2 are the lateral sizes of the structure),

$$\lambda_n \left.\frac{\partial T}{\partial x_3}\right|_{x_3=L_3} = p(x_1, x_2, t) \qquad \text{(A1.3)}$$

for the upper surface, where $p(x_1,x_2,t)$ is the dissipation-density on the surface, and

$$T(x_1,x_2,0,t) = 0 \qquad \text{(A1.4)}$$

for the lowest surface, where ideal heat-sinking is assumed. Additional boundary conditions are valid between the neighboring layers:

$$T(x_1,x_2,s_i\text{-}\varepsilon,t) = T(x_1,x_2,s_i+\varepsilon,t) \quad \text{(i = 1 ... n-1)} \qquad \text{(A1.5)}$$

and

$$\lambda_i \left.\frac{\partial T}{\partial x_3}\right|_{x_3=s_i-\varepsilon} - \lambda_{i+1} \left.\frac{\partial T}{\partial x_3}\right|_{x_3=s_i+\varepsilon} = 0 \quad \text{(i = 1 ... n-1)} \qquad \text{(A1.6)}$$

where

$$s_i = \sum_{k=1}^{i} d_k \qquad \text{(A1.7)}$$

and $\varepsilon \to 0$.
The solution of Eq.(A1.1) will be constructed as a two-dimensional Fourier-cosine series in the two lateral dimensions with coefficients depending exponentially on the third, x_3 dimension. The time dependence is assumed as sinusoidal. For the *i*-th layer

$$T^{i}(x_1,x_2,x_3,t)=\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}\cos(m\pi\frac{x_1}{L_1})\cos(n\pi\frac{x_2}{L_2})\left[A^{i}_{mn}\exp(\gamma_i x_3)+B^{i}_{mn}\exp(-\gamma_i x_3)\right]\exp(j\omega t)$$

(A1.8)

One can prove by substitution that this function, and each summand of the double summation as well, fulfills Eq.(A1.1) if the following condition holds:

$$\gamma_i^2=\left(\frac{m\pi}{L_1}\right)^2+\left(\frac{n\pi}{L_2}\right)^2+j\omega\frac{c_i}{\lambda_i} \qquad \text{(A1.9)}$$

Beyond this, the function (A1.8) fulfills automatically the boundary conditions of Eq.(A1.2) because the expression

$$\cos(m\pi\frac{x_1}{L_1})\cos(n\pi\frac{x_2}{L_2}) \qquad \text{(A1.10)}$$

yields zero at x_1 =0, x_1 =L_1, x_2 =0, x_2 =L_2 for any integer m and n values. The coefficients A^{i}_{mn} and B^{i}_{mn} can be chosen arbitrarily and are suitable to match the solution to the remained boundary conditions of Eq. (A1.3) – (A1.4). For the bottom of the structure

$$A^{1}_{mn}+B^{1}_{mn}=0 \qquad \text{(A1.11)}$$

for the i-th interface

$$A^{i}_{mn}\exp(\gamma_i s_i)+B^{i}_{mn}\exp(-\gamma_i s_i)=A^{i+1}_{mn}\exp(\gamma_{i+1}s_i)+B^{i+1}_{mn}\exp(-\gamma_{i+1}s_i) \qquad \text{(A1.12)}$$

and

$$\gamma_i\lambda_i\left(A^{i}_{mn}\exp(\gamma_i s_i)-B^{i}_{mn}\exp(-\gamma_i s_i)\right)-\gamma_{i+1}\lambda_{i+1}\left(A^{i+1}_{mn}\exp(\gamma_{i+1}s_i)-B^{i+1}_{mn}\exp(-\gamma_{i+1}s_i)\right)=0$$

(A1.13)

For the upper side of the structure

$$\gamma_n\lambda_n\left(A^{n}_{mn}\exp(\gamma_n s_n)-B^{n}_{mn}\exp(-\gamma_n s_n)\right)=P_{mn} \qquad \text{A1.14)}$$

where P_{mn} are the Fourier coefficients of the 2D Fourier expansion of the $p(x_1,x_2)\exp(j\omega t)$ dissipation density function:

$$P_{mn}=\frac{1}{L_1L_2}\frac{4}{(\delta_m+1)(\delta_n+1)}\int_0^{L_1}\int_0^{L_2}p(x_1,x_2)\cos(m\pi\frac{x_1}{L_1})\cos(n\pi\frac{x_2}{L_2})\,dx_2dx_1 \qquad \text{(A1.15)}$$

where δ_i is the Kronecker - δ. (δ_i=1 if i=0 otherwise δ_i=0)
Expressions (A1.11) – (A1.14) constitute a system of linear equations for the 2n unknown A^{i}_{mn} and B^{i}_{mn} values. This system of equations defines a linear relationship between these unknowns and P_{mn} :

$$A_{mn}^{i} = a_i(\lambda_{sp}, \omega) P_{mn} \qquad B_{mn}^{i} = b_i(\lambda_{sp}, \omega) P_{mn} \tag{A1.16}$$

where for a given physical structure a_i and b_i depend on the ω angular frequency and the λ_{sp} wavelength of the spatial harmonics determined by the m and n indices:

$$\frac{1}{\lambda_{sp}^2} = \frac{1}{4}\left(\left(\frac{m}{L_1}\right)^2 + \left(\frac{n}{L_2}\right)^2\right) \tag{A1.17}$$

The process of the thermal field calculation can be thus summarized as follows.

Preparatory step: Using the λ_i , c_i , d_i data of the given structure the functions $a_i(\lambda_{sp},\omega)$ and $b_i(\lambda_{sp},\omega)$ have to be calculated in tabulated form for the λ_{sp} range, that will be used dūring the subsequent calculations. These functions are independent of the dissipation pattern thus are suitable to calculate the thermal field of different surface layout arrangements.

Calculation for a given layout pattern

(i) First the Fourier expansion coefficients of the $p(x_1,x_2)$ surface dissipation density have to be calculated, using Eq. (A1.15).

(ii) The P_{mn} coefficients will be multiplied by the appropriate values of the $a_i(\lambda_{sp},\omega)$ and $b_i(\lambda_{sp},\omega)$ functions according to Eq. (A1.16).

(iii) Eq. (A1.8) can be used to obtain the temperature field of any layer.

Step (i) involves a 2D Fourier expansion step, (iii) means a Fourier reconstruction step. The number of elements in the Fourier expansion should be limited owing to practical reasons. This number determines the spatial resolution of the calculation. Considering a chip of 2mm by 2mm size the solution using 1024 by 1024 elements Fourier series gives a resolution of about 2μm.

In the practical realization of the algorithm we work with sampled versions for both the $p(x_1,x_2)$ power dissipation and $T(x_1,x_2,x_3\!=\!s_n)$ surface temperature functions. Therefore the need of a pre-sampling, anti-aliasing spatial filtering is usually raised.

On the sampled 2D functions discrete Fourier transformation (DFT) has to be executed. For this goal the most convenient way is the use of the Fast Fourier Transformation (FFT) method. This algorithm provides a very fast computing which is essentially the main advantage and the primary attractive property of the Fourier method.

Free convection cooling of the surface can take an important part in the thermal effects. The air around the investigated structure has very bad heat conduction properties, however, the amount of heat transfer through that media can not be neglected in some cases. This phenomena can be approximated with the THERMANAL by adding one or more pseudo-layers to the layer structure at the bottom or/and at the top of it. This pseudo-layers should model the additional heat transfer and heat conduction to the ambient. This method is equivalent with the use of the *heat transfer coefficient* calculation of the natural convection, which is an acceptable model in still air.

A1.2. Extension for bulk dissipation

The assumption that the dissipation arises exactly on the surface is a rather rough one in many cases. The conventional ICs are usually covered by a protection layer. Moreover the active region of the dissipating components (as e.g. the collector-base junction of a bipolar transistor) are situated in some depth from the surface. Stacked 3D packaging of IC's (3D Multi-Chip modules) is constituted of multi-layered blocks as well, with dissipating areas on the layer interfaces, that is on the surfaces of the individual layers.

A slight extension of the algorithm described above allows us to solve such problems as well. Let us complete the model in such a way that dissipating elements lay not only on the very top surface of the block but on the interface plane between layers, too. Let us denote the dissipation density between the *i*-th and *i*+1 th layers by $p_i(x_1,x_2)$. Only the boundary condition of Eq. (A1.6) has to be changed to the form of

$$\lambda_i \frac{\partial T}{\partial x_3}\bigg|_{x_3=s_i-\varepsilon} - \lambda_{i+1} \frac{\partial T}{\partial x_3}\bigg|_{x_3=s_i+\varepsilon} = p_i(x_1,x_2) \qquad \text{(i = 1 ... n-1)} \qquad \text{(A1.18)}$$

The corresponding equation in the Fourier algorithm which will substitute Eq. (A1.13) is

$$\gamma_i\lambda_i\left(A^i_{mn}\exp(\gamma_i s_i) - B^i_{mn}\exp(-\gamma_i s_i)\right) - \gamma_{i+1}\lambda_{i+1}\left(A^{i+1}_{mn}\exp(\gamma_{i+1}s_i) - B^{i+1}_{mn}\exp(-\gamma_{i+1}s_i)\right) = P^i_{mn}$$

$$\text{(i=1... n-1)} \qquad \text{(A1.19)}$$

where P^i_{mn} are the Fourier coefficients of $p_i(x_1,x_2)$. In this case the linear system of Eqs. (A1.11), (A1.12), (A1.19) and (A1.14) results in the following linear relationships

$$A^i_{mn} = a_{ij}(\lambda_{sp},\omega)P^j_{mn} \qquad B^i_{mn} = b_{ij}(\lambda_{sp},\omega)P^j_{mn} \qquad \text{(A1.20)}$$

As it can be seen the number of Fourier transformation steps during the simulation has increased. The coefficients P^j_{mn} have to be determined by 2D Fourier expansion for all *j*, so the 2D Fourier expansion has to be repeated *k* times, where k is the number of dissipation planes in the actual task. After the calculation of the A^i_{mn} and B^i_{mn} coefficients, the temperature distribution in any plane can be obtained by 2D inverse Fourier transformation, so the number of 2D inverse Fourier transformation steps equals to the number of the investigated planes (chosen by the user). All the other features of the algorithm remain unchanged.

A1.3. Extension for heat transfer via beam-leads

There are microelectronic parts suspended on thin and narrow strips. Beam-lead packages belong to this class as well as the ball-grid array attachment of conventional IC chips. The Fourier algorithm can be extended for these cases as well supposing that

- **(i)**the suspended structure is also a multi-layered rectangular one,
- **(ii)**one or more necks or beam-leads that support the structure are narrow enough to be treated as one-dimensional from the point of view of heat transfer.

The leads act as heat-sinking elements. Heat sinking elements can be treated as dissipating elements with negative dissipation. If we are able to calculate this negative dissipation for each lead the solution of the structure can be obtained using the algorithm described in Section A1.1. These negative dissipation values can be obtained as follows.

The attaching points of the leads are considered as ports of a linear thermal network. When we have k leads the chip can be treated as a linear k -port. First we suppose that the leads are disconnected and calculate the heat distribution. Let us denote the temperatures of the lead connection points by T_{io}. In the second step we calculate the thermal impedance matrix ZC_{ij} of the linear k -port. This matrix can be calculated in the following way. First we assume a unit-power dissipation to be forced on the 1st port while all other dissipating elements are disabled. After solving the problem and obtaining the temperature distribution the temperatures of the 1st... k th ports give the first column of the ZC_{ij} matrix. In a similar manner all the columns of the matrix can be obtained.

Let ZL_{ij} be the heat resistance (impedance) diagonal matrix of the leads. For one-dimensional leads with constant cross-section this can be calculated by analytical equations. Using the elementary methods of network theory for multi-ports we obtain:

$$T_{io} = (ZC_{ij} + ZL_{ij})P_j \qquad \text{(A1.21)}$$

Solving this set of equations for P_j we obtain the vector of heat flux flowing through the leads. Taking into account these fluxes as negative dissipations on the lead connection points we can get by a final solution step the true temperature map of the lead-suspended structure.

References to A1

[1] A.G.Kokkas: Thermal analysis of multiple-layer structures, *IEEE Trans. on El.Dev.*, V.ED-21, No. 11, pp.674-681, 1974

[1] V. Székely, P.Baji, M.Rencz: Graphical computer methods in the design of integrated circuits, Periodica Polytechnica, V.23, No.3-4, pp 331-338, 1979

[2] V. Székely, A. Csendes, M. Rencz: μS-THERMANAL: an efficient thermal simulation tool for microsystem elements and MCM's, *SPIE'96 Symposium on Micromachining and Microfabrication,* 14-15 Oct. 1996, Austin, Texas USA, SPIE Proc. V.2880, pp.64-75.

Appendix 2.

Algorithm of the SUNRED program

A2.1. The 2D model

The 2D version of the program treats the linear heat conduction problems in two dimensions. Anisotropy can be taken into account. The equation being solved is

$$p(x,y)+c\frac{\partial T}{\partial t}=\frac{\partial}{\partial x}\left(\lambda\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(\lambda\frac{\partial T}{\partial y}\right) \qquad \text{(A2.1)}$$

and in the steady-state case

$$p(x,y)=\frac{\partial}{\partial x}\left(\lambda\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(\lambda\frac{\partial T}{\partial y}\right) \qquad \text{(A2.2)}$$

This is the 2D form of the well-known Poisson equation, the mathematical description of many physical phenomena.

The investigated area is a rectangle. A dense equidistant grid is spawned to this area defining a cell matrix. The suggested grid size is either 256×256 or greater. A material type is assigned to each cell. Constructing an image – in the sense of the digital image handling methods performs this assignment. Each pixel of this digital image corresponds to a grid-cell whereas the material type constituting the cell is coded by the color of the pixel.

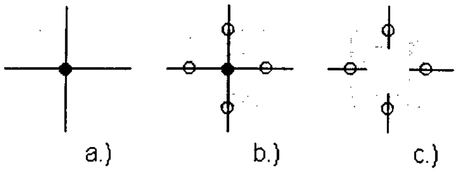
Thus, in order to enter a problem, two files have to be prepared:

- a "problem-image" which can be in any usual image format,
- a "material-table" assigning different material parameters to each color.

This method of problem definition provides a very easy and fast input of complex geometrical arrangements (using any general picture editing tools). Almost arbitrarily shaped structures can be investigated; limitation is coming only from the finite resolution of the digital image.

On the edges of the investigated rectangular area either forced temperature or zero heat-flow can be prescribed – individually, for any grid points of the boundary. Excitations can be defined in the interior of the investigated area as well, forcing a given temperature or a given heat-flux to any cell. Obviously a new "color" should be introduced for each excitation value in the problem image.

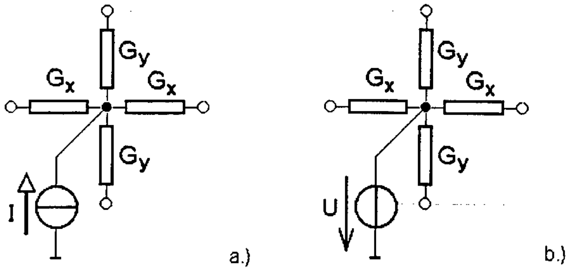
The solution of Eq. (A2.1) is accomplished by using the method of *finite differences*, and applying a network model for the thermal field. An electrical model describes the cells of the field. The cells are squares (or rectangles), with a node in their center (Fig.A2.1a.). Heat flux can be forced into them – this corresponds to the current flowing in this node. Forced temperature means the forced value of the cell node.



• Fig.A2.1. Cell, center node and terminal nodes

The boundary between different materials is lying always on the cell edges. In other words: each cell is "filled" by a single material. Each cell has four terminals in the direction of its four neighbors (Fig.A2.1b.). On the terminals each cell can be described by a 4×4 matrix. This way the center node is hidden, but knowing the terminal temperatures the temperature of the center node can be back calculated. Fig.A2.1c. presents that the cell shows four terminals to the outside and the inner node is hidden.

The steady-state model of the cell is shown in Fig.A2.2. It contains four thermal conductances. The value of these conductances depends on the thermal conductivity of the material filling the cell and on the geometry. This basic cell can be described by an admittance matrix of 4×4 size.



• Fig.A2.2. Steady state circuit models of a single cell.

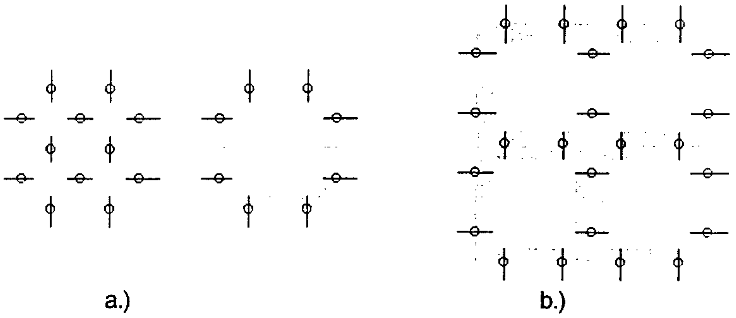
a.) Current excitation, b.) forced voltage

A2.2. The solution algorithm [1]

The solution of the problem is done by the electrical solution of the whole model network. This raises serious problems because of the size of this network. For a 256×256 grid arrangement the model network consists of 131072 nodes. Although the corresponding circuit matrix is extremely sparse the solution of such a big network is a hard problem.

In order to avoid the troublesome "when to finish the iteration" problems we have not considered iterative solutions – only direct methods have been investigated. A successive procedure has been developed for the network reduction. The essential features of this algorithm are briefly presented in this paragraph.

Four basic cells can be assembled to form a block or macrocell as shown in Fig.A2.3.a. In other words: a *1st order cell* has been built from four *zero-order cells*. The four interior connecting terminals of the cells can be eliminated; they will not appear in the outside-directed description.



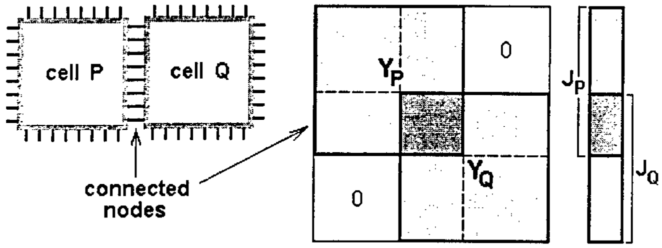
- Fig.A2.3. Network reduction.

- a.) Four basic cells will constitute a 1st level cell, b.) Building a 2nd level cell

Using four 1st level cells we can assemble a 2nd level cell as shown in Fig.A2.3b. The inner terminals can be eliminated again.

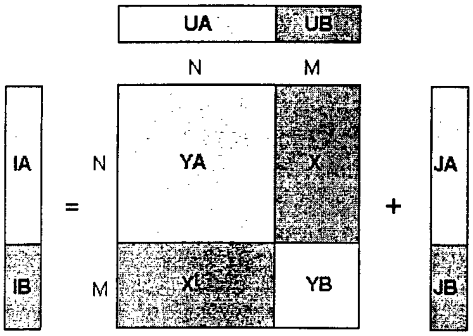
Continuing this successive construction of higher and higher level cells we obtain finally the matrix of a single cell – the terminals of which are lying on the four edges of the investigated rectangular field. Matching with the boundary conditions means the solution of this matrix for the U or I constraints, given individually for the terminals lying on the boundaries of the investigated field. The voltages of all the inside nodes can then be calculated by a successive back-substitution.

Let us present the procedure in terms of the data flow and arithmetic operations used. The cells are described by their admittance matrices **Y** relating to the boundary nodes and by their inhomogeneous vectors **J** representing the excitations on all the inside nodes but reduced to the boundary nodes. For the 0th level cells shown in Fig.A2.1. **Using elementary calculations can generate Y and J**. The connection of two cells as shown in Fig.A2.3. is equivalent to the addition of their **Y** matrices and **J** vectors where the areas of the connected nodes overlap as shown in Fig.A2.4. as well.



• Fig.A 2.4. Connection of two cells and the resulting Y matrix and J vector

The next step is to eliminate the inside (≡connected) nodes. For sake of better understanding **Y** and **J** are visualized in Fig.A2.5. in a rearranged node order. The first *N* nodes are the boundary nodes (that should be kept), *M* are the inside nodes that being eliminated. Partitions of **Y** are denoted by **YA**, **YB**, **X** and **X**^{t} as shown in Fig.A2.5.



• Fig.A2.5. Partitions of the admittance matrix

The nodal voltages and nodal currents are represented by the **U** and **I** vectors, respectively. The **J**, **U** and **I** vectors are partitioned similarly to **Y**. The linear matrix equation for the two connected cells is

$$\mathbf{IA} = \mathbf{YA} \cdot \mathbf{UA} + \mathbf{X} \cdot \mathbf{UB} + \mathbf{JA} \qquad (A2.3)$$

$$\mathbf{IB} = \mathbf{X}^t \cdot \mathbf{UA} + \mathbf{YB} \cdot \mathbf{UB} + \mathbf{JB} \qquad (A2.4)$$

Elementary rearrangements of these equations result in the formula for the reduced admittance matrix

$$\mathbf{Y}_{\mathrm{RED}} = \mathbf{YA} - \mathbf{X}\cdot\mathbf{ZB}\cdot\mathbf{X}^{t} \qquad \text{(A2.5)}$$

where **ZB=YB**$^{-1}$. The new inhomogeneous part is

$$\mathbf{J}_{\mathrm{RED}} = \mathbf{JA} - \mathbf{X}\cdot\mathbf{ZB}\cdot\mathbf{JB} \qquad \text{(A2.6)}$$

During the back-substitution step **[UA]** → **[UA, UB]**

$$\mathbf{UB} = -\mathbf{ZB}\cdot\mathbf{X}^{t}\cdot\mathbf{UA} - \mathbf{ZB}\cdot\mathbf{JB}\ . \qquad \text{(A2.7)}$$

It is worthy for note that the same matrix is appearing in (A2.6) and (A2.7) because

$$\mathbf{ZB}\cdot\mathbf{X}^{t} = (\mathbf{X}\cdot\mathbf{ZB})^{t}\ . \qquad \text{(A2.8)}$$

All the **Y** matrices are symmetrical. This permits a saving of about 50% both in storage and in arithmetic operations.

The description of all cells by their **Y** matrices represents a huge amount of data. As the processing is essentially serial, it is advantageous to store these data streams in files. Thus the organization of the program is mainly *pipelined*: the program segments read one or more streams from files and writes the results into further files. This way a quite large number of nodes can be handled on computers having only limited amount of memory.

Organization of the operations

The short description of the main program segments demonstrates clearly the pipelined process:

1. Network reduction. The segment reads the queue of the **Y** matrices of *n*th level cells, reduces them in fours by using three times Eq. (6) and writes the resulting, *n*+1th level **Y** matrices into a new file. The **ZB** and **X·ZB** matrices are calculated and stored into a further file as well. The number of runs of this segment is $\log_2(K)$, where *K* is the number of the pixels in one edge of the problem-image.

2. Forward substitution. This segment reads the file of the **X·ZB** matrices, reads the queue of the **J** inhomogeneous vectors of *n*th level cells, reduces them in fours by using Eq. (7) and writes the resulted, *n*+1th level **J** vectors into a new file. The number of required runs is $\log_2(K)$ again.

3. Solution. This segment uses the uppermost level **Y** matrix and **J** vector and solves the corresponding system of linear equations, taking into account the actual boundary parameters.

4. Backward substitution. This segment calculates the voltages on the internal nodes in a hierarchical top-down order, by using Eq. (8) and two files: the queues of the **J** vectors and the **X·ZB** matrices.

The advantage of this ordering of the calculus lies in the fact that the first and most time consuming step has to be repeated only in the case when the investigated structure has been changed. When the excitations are only changed, steps 2, 3 and 4 have to be repeated. In the case when only the boundary conditions are modified repeating of the steps 3 and 4 is sufficient.

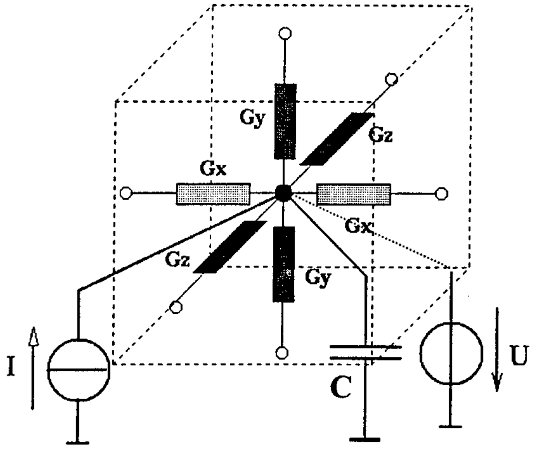
A 2.3. Extension of the SUNRED algorithm for 3D

The 3D model

In the model of the 3D SUNRED program the investigated volume is an orthogonal parallelepiped. A dense mesh is spawned to the volume defining a three-dimensional cell matrix. A single material property is assigned to each cell. In the *x* and *y* dimensions the resolution (the number of the cells) must be the same while in the *z* direction it can be different. This enables simplified handling in the case of layered structures where the resolution in the third dimension can be reduced if the layers are of homogeneous material. Such (so called 2 and half dimension) structures are quite usual in microelectronics and their treatment in this way requires much less computation than the usual 3D simulation.

The definition of the problem to be analyzed is a basic question in case of 3D field solvers. In the case of the 2D SUNRED program the innovative problem definition treats the structures to be analyzed as images. In the present version of the 3D SUNRED program the two dimensional problem description is extended to handle the layered structure where each pixel in a single layer corresponds to a grid cell. This assures that real images like printed circuit board masks may be used as geometry input. It means that, the problem geometry is defined with the help of layers given by a series of images. The cells and the layer thickness describe the layers. In the case of the SUNRED program the mesh is not restricted to be equidistant. A non-uniform mesh can be prescribed for the cells, which further reduces the computation time where the grid resolution is not so critical.

The Poisson equation for the structure is constructed by using the finite difference method applying a network model for the thermal field. An *electrical model* describes the cells with a node in the center. Each cell has six terminals in the direction to its six neighbors. On the terminal points their admittance matrices can describe each cell. If we know the temperature of the terminal points the temperature of the center node can be back calculated. The values of the six thermal conductances' in each cell can be calculated from the material properties and of the size of the cell



• Figure A 2.6 Circuit model of the single cell.

On the edges of the investigated rectangular area either forced temperatures or zero heat-flow can be prescribed. Excitations can be defined in the interior of the investigated area as well, either by forcing a given temperature (*U* constraint) or a given heat-flux (*I* constraint) to any cell, see Fig. A.2.6.

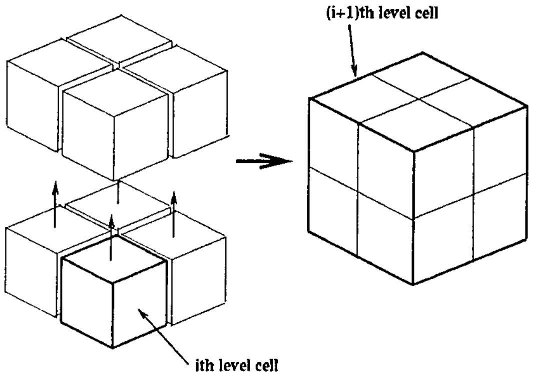
In the case of *the transient solution* a capacitance is added to each cell between the center node and the ground, the value of which is determined by the heat capacitance of the cell.

In the case of calculating *in the frequency domain* the model is the same, but the elements are frequency dependent. Even the solution algorithm can be the same, except that the elements of the admittance matrix are complex numbers.

The 3D solution algorithm

The solution of the problem is done by the electrical solution of the model network. The solution of the three dimensional problem is similar to the two-dimensional case, but the resulting matrices are now even much larger. For the theoretical background of the successive network reduction algorithm refer to [3].

In the three-dimensional case eight basic cells are assembled to form a higher-level macro cell. This means, that a *1st order cell* is built from eight *zero order* cells. The 12 interior connecting terminals of the cells can be eliminated, they will not appear in the outside-directed description. Using eight 1st level cells we can assemble a 2nd level cell etc. (See Fig. A.2.7). The inner terminals can be eliminated again.



• Figure A.2.7. Cell assembly.

Continuing this successive construction of higher level cells we obtain finally the matrix of a single cell – the terminals of which are lying on the six sides of the investigated orthogonal parallelepiped.

The solution of the problem can be calculated by matching the prescribed boundary conditions to the top-level cell and using back substitution steps to calculate the results for each internal node.

The cells are described by their admittance matrices **Y** relating to the boundary nodes and by their inhomogeneous vectors **J** representing the excitations on all the inside nodes but reduced to the boundary nodes. For the 0th level cells shown in Figure A.2.6 elementary calculations can generate **J** and **Y**. The connection of two cells is equivalent to the addition of their **Y** matrices and **J** vectors where the areas of the connected nodes overlap.

The next step is to eliminate the inside (≡connected) nodes. **Y** and **J** are shown in Fig.A.2.5. in a rearranged node order. The first *N* nodes are the boundary nodes (that should be kept), *M* are the inside nodes that will be eliminated. Partitions of **Y** are denoted by **YA**, **YB**, **X** and **X**t.

The vectors **U** and **I** represent the nodal voltages and nodal currents. The **J**, **U** and **I** vectors are partitioned similarly to **Y**. The linear matrix equation for the two connected cells is, similarly to Eq.s (A2.3) –(A2.8)

$$\mathbf{IA} = \mathbf{YA} \cdot \mathbf{UA} + \mathbf{X} \cdot \mathbf{UB} + \mathbf{JA}$$

$$\mathbf{IB} = \mathbf{X}^t \cdot \mathbf{UA} + \mathbf{YB} \cdot \mathbf{UB} + \mathbf{JB}$$

Elementary rearrangements of these equations result in the formula for the reduced admittance matrix

$$\mathbf{Y}_{\mathbf{RED}} = \mathbf{YA} - \mathbf{X} \cdot \mathbf{ZB} \cdot \mathbf{X}^t$$

where **ZB=YB**$^{-1}$. The new inhomogeneous part is

$$\mathbf{J}_{\mathbf{RED}} = \mathbf{JA} - \mathbf{X}\cdot\mathbf{ZB}\cdot\mathbf{JB}$$

During the back-substitution step **[UA]** → **[UA,UB]**

$$\mathbf{UB} = -\mathbf{ZB}\cdot\mathbf{X}^t\cdot\mathbf{UA} - \mathbf{ZB}\cdot\mathbf{JB}$$

$\mathbf{ZB}\cdot\mathbf{X}^t = (\mathbf{X}\cdot\mathbf{ZB})^t$ are the same matrices, just like in the 2D case.

The three-dimensional implementation exploits the hierarchical nature of the network reduction and organizes the data in the same pipelined manner as in the two dimensional implementation.

An interesting property of the algorithm is that the describing admittance matrices can be constructed independently, so it is possible to obtain them by using parallel calculation.

The original 2D implementation handles the admittance matrices of the problem as upper triangular matrices. However, the admittance matrices have a unique property, namely that they are symmetric positive definite matrices. This enables to use efficient factorization methods, which reduces the computation time again. In the present version the algorithm uses a special form of the Cholesky-factorization to invert the matrices during the successive reduction and the boundary solution.

Efficiency

The operation requirement of a single matrix inversion is Ordo(P^3) in general cases where P is the number of nodes in the structure.
Sparsity can be exploited by using different methods. The successive node reduction gives the following numbers for the necessary floating point operations:

$$N_{2D} \approx 127P^{1.5}$$
$$N_{3D} \approx 216P^{2}$$

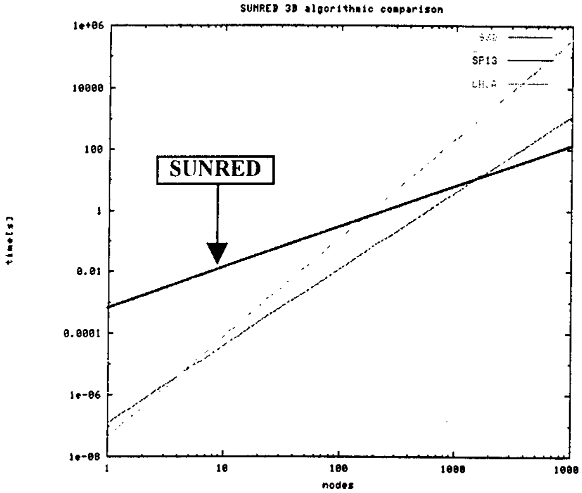
where P is the number of nodes in the whole network. The most important result is that the 3D SUNRED algorithm requires Ordo(P^2) operations to calculate the results.

Up to now the SUNRED 3D algorithm has been compared with four very important traditional sparse techniques:

- the solver described by Chua in [4].

- the Sparse 1.3 library from the Univ. of Berkeley, USA in
- the iterative sparse symmetric Gauss-Seidel method [2].
- the iterative sparse symmetric successive over-relaxation method [2].

The comparison was done on a Sun Enterprise 2170 computer and the results are given in Fig. A.2.8. The computational time is considered as the sum of the factorization time and the solution time. The comparison demonstrates that the SUNRED 3D program is faster than any other of the examined, extensively used methods, if the number of nodes is greater than ≈ 2000. According to our detailed study this is because SUNRED exploits the special structure of the analyzed problem and spends less time with the reordering of the equations.



• Figure A.2.8 Comparison of SUNRED (S3D) with traditional methods

Changing the dimension from two to three results in a computationally harder problem: A 64 x 64 grid in 2D contains the same number of cells as a 16 x 16 x 16 grid in 3D. Whereas the size of the top-level matrix for this 2D problem is only 256 x 256, but for the 3D case it is already 1536 x 1536. This means that the time consumed by the boundary solution is about 200 times longer for this same element-number problem in 3D than in the 2D case.

The gain in the computation time over the other sparse matrix solver algorithms is slightly less than in the case of 2D SUNRED, although still considerable, especially in the case of large node numbers. The not so high gain is originated from the large number of matrix elements: the algorithm spends more than 75 percent of the computation time with matrix multiplication operations.

References to A2

[1] V. Székely, M. Rencz, „Fast field solver programs for thermal and electrostatic analysis of microsystem elements", in the Proceedings of ICCAD'97, Santa Jose, CA, USA, 9-13 Nov 1997, pp. 684-689.

[2] Barnett et al., „Templates for the solution of linear systems: Building blocks for iterative methods", DARPA & ARO supported project DAA L03-91-C-0047, 1991

[3] V. Székely, M. Rencz, „Fast field solvers for thermal and electrostatic analysis", in Proc. of DATE'98, Paris, France, 23-26 Feb 1998, pp. 518-523.

[4] L. O. Chua, Pen-Min Lin, „Computer-aided analysis of electronic circuits", Prentice-Hall, Englewood Cliffs, 1975.

[5] http://www.netlib.org/sparse, „Sparse 1.3"

[6] Framasoft+CSI, „Systus User's Manual", Tour Fiat, Cedec 16 92084, Paris La Defense, France, 1995.

[7] V. Székely, A.Pahi, M. Rencz: "SUNRED, a new field solving approach" accepted for publication at the SPIE conference of CAD, Design and Test, 30 March- 1.April 1999, Paris, France

Algorithm of the THERMODEL program

A.3.1. Theoretical background

The theoretical background of the model generation method used in the THERMODEL program is based on a new representation of the distributed RC networks. In this representation the behaviour of the network is described by convolution equations. Here we present this theory very briefly, limited only to the elements needed to follow the algorithm of the THERMODEL tool.

In the following calculation the *t* time and the ω angular frequency will be substituted by their natural logarithm:

$$z = \ln(t) \qquad \Omega = \ln(\omega) \tag{A3.1}$$

The *R*(*z*) *time-constant density* function will be defined in order to represent the RC circuits either in lumped-element or in distributed circuit case. This function gives the intensity of the terms of different time-constants in the response. We can interpret this function as a special kind of a spectrum, which depicts the occurrence and the relative intensity of the different time-constants in the circuit response. Sometimes we will use the alias-name *time-constant spectrum* as well.

The *R*(*z*) function is a sum of Dirac-δ's in case of a lumped element network where the response consists of terms of discrete time-constants in a finite number:

$$R(z) = \sum_{i=1}^{p} K_i \, \delta(z - \ln \tau_i)) \tag{A3.2}$$

τ_i are the time-constants of the poles, K_i are the corresponding magnitudes, *p* is the number of the poles. The *R*(*z*) spectrum is a continuous function in case of an infinite distributed network. In this case the definition is

$$R(z) = \lim_{\delta z \to 0} \frac{magnitudes\ relating\ time - cons\tan ts\ between\ z\ and\ z + \delta z}{\delta z} \tag{A3.3}$$

The *R*(*z*) time-constant spectrum is related both to the time and the frequency responses. The relation is of convolution-type in both cases. In the time-domain

$$\frac{d}{dz}a(z) = R(z) \otimes w_t(z) \qquad \text{(A3.4)}$$

where *a*(*z*) is the unit-step response, ⊗ is the convolution operator and the $w_t(z)$ function is defined by

$$w_t(z) = \exp(z - \exp(z)) \qquad \text{(A3.5)}$$

In the frequency-domain, for the **Z**(ω) impedance function the following equation can be applied:

$$-\frac{d}{d\Omega}\mathrm{Re}(\boldsymbol{Z}(\Omega)) = R(z = -\Omega) \otimes w_r(\Omega) \qquad \text{(A3.6)}$$

where

$$w_r(\Omega) = 2 \cdot \frac{\exp(2\Omega)}{(1 + \exp(2\Omega))^2} \qquad \text{(A3.7)}$$

Both convolution equations are suitable for the model identification. If we know the time-response of the network Eq.(A3.4) should be used. Knowing the frequency-domain behaviour Eq.(A3.6) has to be applied. In either case the inverse operation of the convolution: the *deconvolution* should be accomplished to extract the *R*(*z*) time constant spectrum from the network responses.

Unfortunately we do not have a straightforward way to execute the deconvolution. In most cases this operation is extremely ill-defined, and as a consequence the smallest inaccuracy or noise in the input function (in the network response in our case) makes the result completely useless. There are various methods to overcome these difficulties and the solutions are usually tailored to the individual problems.

A usual solution is the *Fourier-domain inverse filtering*. As it was shown previously the responses of the network are obtained by convolution integrals, as

$$m(x) = R(x) \otimes w(x) \qquad \text{(A3.8)}$$

– where *m*(*x*) is a response of the network, *R*(*x*) is the time-constant spectrum and *w*(*x*) is one of the two weighting-functions of (A3.5) and (A3.7). Turning into the Fourier domain we have the corresponding formula as

$$M(\Phi) = R(\Phi) \cdot W(\Phi) \qquad \text{(A3.9)}$$

where the capital letters represent the Fourier transform functions whereas Φ is their "frequency" variable. This equation contains multiplication instead of convolution.

It must be emphasized that the $m(x) \rightarrow M(\Phi)$ transformation is not the same as the usual transformation into the frequency-domain since the variable x is not the time but ln(*time*) or ln(*frequency*). The frequency Φ can be interpreted as the number of waves per frequency-decade or time-decade on the logarithmic $R(z)$ function.

The Fourier domain, the space of the Φ frequencies is well suited to execute the deconvolution (the inverse of convolution) since (A3.9) yields

$$R(\Phi) = \frac{M(\Phi)}{W(\Phi)} \qquad \text{(A3.10)}$$

Thus theoretically we can obtain the required function by a simple division. Since the higher Φ frequency components of $W(\Phi)$ are quite small the division enhances the higher frequency values of $M(\Phi)$$ extremely, resulting in an unwanted enhancement of the noise. This enhanced high-frequency noise can be as large as or even larger than the useful part of the function and can completely hide them. This fact constitutes the ultimate limit of the resolution or accuracy of the deconvolution.

The relation between the noise level and the resolution limit of the deconvolution is detailed in a recent work [] for the network identification problem. Here we recall only some results of these investigations. Let us examine for instance the identification using Eq.(A3.6). If we have $m(x)$ with an accuracy of 10^{-8} (which is not impossible in case of a response produced by simulation) the possible resolution of the approximate $R(z)$ function is 0.66 octave. This means that a single line of a discrete-line spectrum is broadening to a finite-width peak the half-value width of which is approx. 0.66 octave.

The last problem to be solved is to find a procedure suitable to generate a lumped element equivalent network in the knowledge of the formerly determined $R(z)$ function.

The $R(z)$ function of the distributed networks is continuous if their length can be taken to be infinite. Owing to the finite resolution, identification of lumped networks results in continuous $R(z)$ functions as well. In order to build lumped models we have to approximate these continuous functions by a set of discrete spectrum lines. This approximation always involves a trade-off. On one hand, it is practical to keep the number of these lines as low as possible in order to minimize the size of the model network. On the other hand the error of the approximation should remain below an allowable limit.

Let us discuss the time-constant density function in the case when $R(z) \geq 0$ (case of driving point impedances). A possible approach is the direct discretization of the $R(z)$ function. The most straightforward way for this is the equidistant placement of the poles on the logarithmic *z* or Ω axis. If the relevant part of the function is in the $[z_a, z_b]$ region of the *z* axis (see Fig.A3.1) then the location of the poles is given by

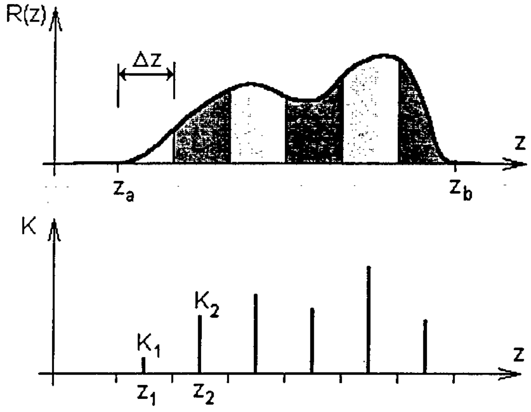
$$z_i = z_a + \left(i - \frac{1}{2}\right) \cdot \Delta x \qquad i=1\ldots N \qquad \text{(A3.11)}$$

where *N* is the pole number of the approximation, and

$$\Delta z = \frac{z_b - z_a}{N} \qquad \text{(A3.12)}$$

The magnitudes of the discrete spectrum lines can be calculated by the following simple expression:

$$K_i = \int_{z_i - \Delta z/2}^{z_i + \Delta z/2} R(\varsigma) d\varsigma \qquad \text{(A3.13)}$$



• Fig.A3.1. Discretization of the time-constant spectrum

Possessing now the discrete-line *R*(*z*) spectrum the construction of the RC-ladder model network is a routine task of the linear network theory.

In the case of transfer functions a special problem appears typically: the magnitudes are partly negative. An easy way to cope with this problem is to realize the positive and the negative part of *R*(*z*) separately, by two distinct RC one-ports and finally to add the output variables of these two sub-networks with the appropriate sign.

References to A3

[1] D.Gardner, J.Gardner, G.Lush, W.Meinke, "Method for the analysis of multicomponent exponential decay curves", J.Chem. Phys. 31,978-986 (1959)

[2] V.Székely, M.Rencz, B.Courtois: Thermal Transient Testing, Microelectronics International, No 43, May 1997

[3] V.Székely: A new Evaluation Method of Thermal Transient Measurement results, Microelectronics Journal 28 (1997) 277-292

[4] V.Székely, M.Rencz, B.Courtois: A Step Forward in the Transient Thermal Characterization of Packages, 1997 Proceedings, International Symposium on Microelectronics, pp 296-301

[5] V. Székely: Identification of RC Networks by Deconvolution: Chances and Limits, IEEE Transaction on Circuits and Systems-I. Theory and Applications, Vol.45, No.3, March 1998, pp 244-258